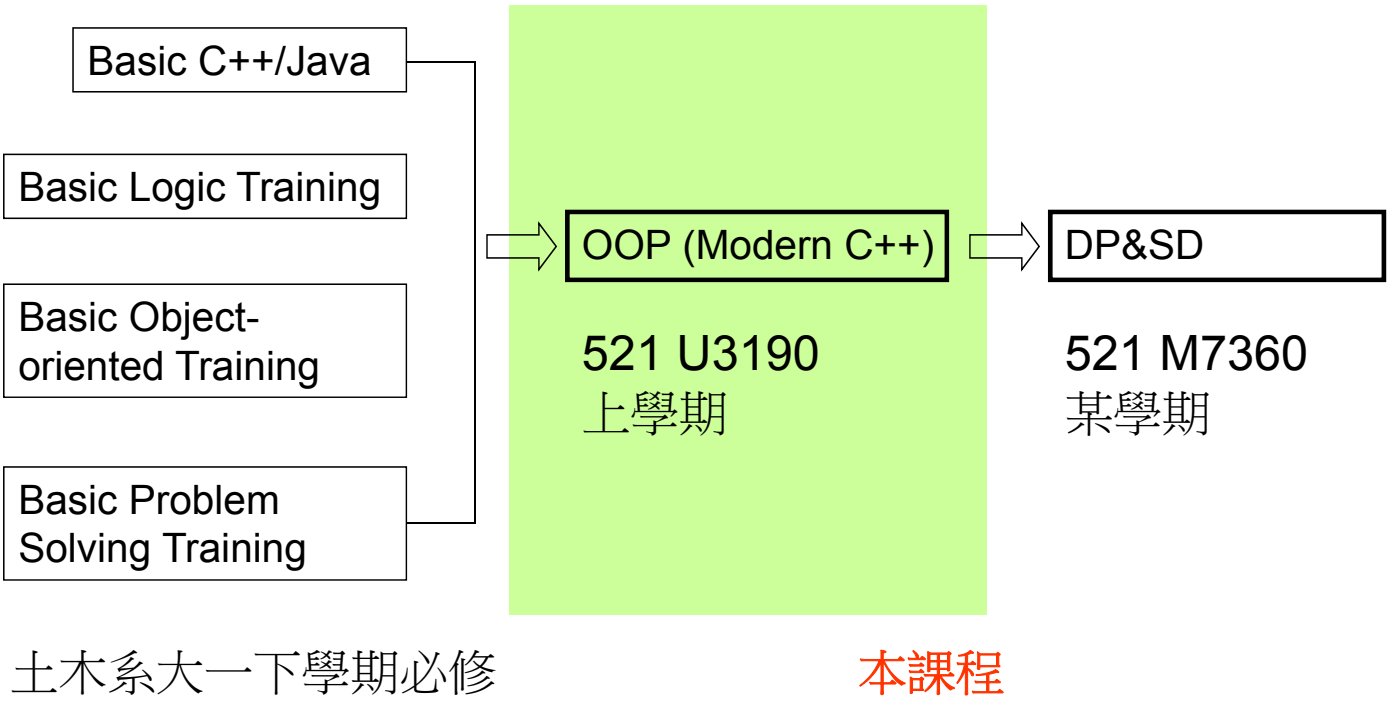
521 U3190 物件導向程式語言
Object-Oriented Programming Language

- Welcome!
- 授課老師：陳俊杉 dchen@ntu.edu.tw
- 助教：林欣瑞、林致淳、林冠伯、吳若凡
- 課程網站
  - ceiba.ntu.edu.tw
- This is a **demanding** course and **SHALL NOT** be your first programming course!
- 修課人數上限 **55** 人 (修課優先順序與抽籤原則)
- 歡迎旁聽 (不過Lab以修課同學為優先)
- Syllabus (see note)

Course Roadmap: 三部曲



課程目的

- Learn the whole world of **modern C++** including its object-oriented and non-object-oriented features.

Modern C++

- The low-level language, largely inherited from C
- More advanced language features (class, template) that allow us to define our own data types and to organize large-scale programs and systems
- The standard library (STL), which uses these advanced features to provide a set of useful data structures and algorithms
- **C++11** (2011 C++ Standard)

Why learning programming language?

- You learn it because it is fun!
- You learn programming language so you can **express your ideas** to **accomplish tasks with computers**.
- Our civilization runs on software and **programming is a way to reach out and change the world**.

我很早便在蘋果觀察到一件事，我常常想到，但不知道該如何解釋。

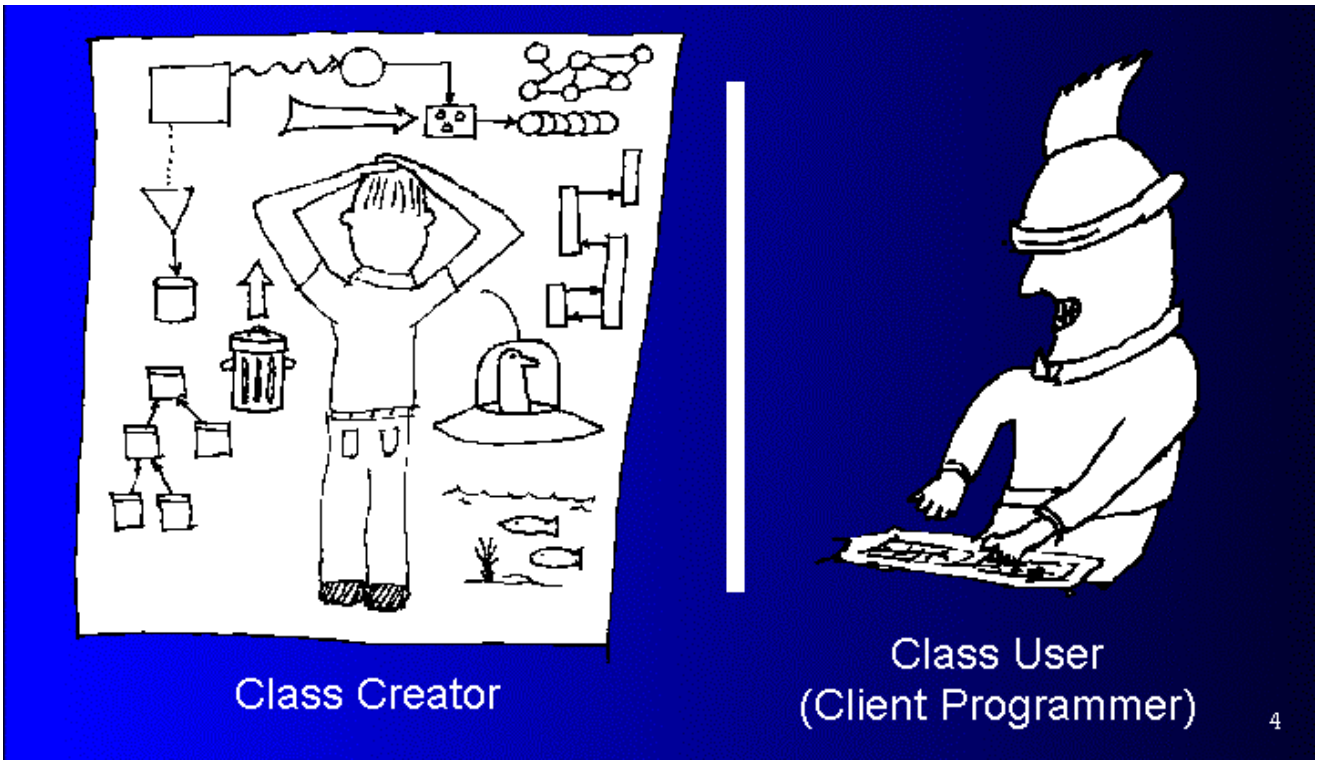
人生中大多數事情，平庸與頂尖的差距，通常差距只有二比一，假如在紐約搭上一般司機的車，與最棒的司機比，最棒的司機也許能讓你快三〇%的時間到達目的地。

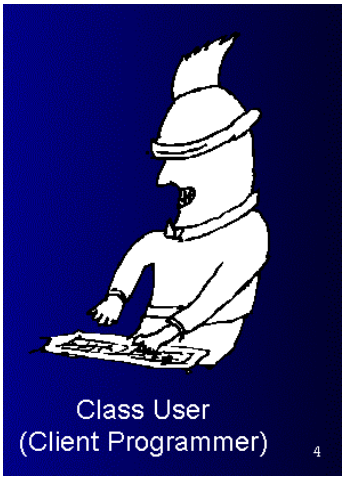
普通汽車和頂尖汽車的差異有多少？也許二〇%吧。頂級CD播放機和一般CD播放機的差別？我不知道，也許二〇%吧。因此，二比一在人生中已經是極大差異。但是，就軟體而言，平庸和頂尖的差異，可能達五十比一，甚至是一百比一，這種情況在生活中很少見，我很幸運的能把我的人生，花在這樣的領域上。

Why Learning C++?

- C++ supports key concepts and techniques used in real world applications.
- C++ poses the balance between elegancy and efficiency.
- C++ programming concepts can be used directly in other languages (C, C#, Fortran and Java).

Modern C++ 教學原則與方法

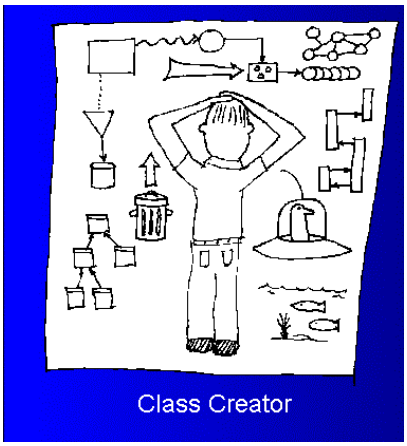




- Basics
- C++ Library (IO, Containers, Algorithms, Dynamic Memory)

Learning Objectives

- Learn how to write C++ programs
- Learn how to use the abstractions from the library
- Imitate STL and understand how to build your own types



- Tools for Class Authors (Class, Operator Overloading, Object-Oriented, Template and Generic Programming)

Learning Objectives

- Learn how to write your own **types** in C++.

Philosophical Note on Learning C++

- You need to understand the principles.
- You need to have your hands dirty to fully appreciate it.
- Skills come with practice!

本課程理想的進行方式 (I)

透過課前預習、上課互動式抽問、重點解釋等方式加深學習效果

- (我) 每次上課結束前會告知下星期的進度
- (我) 會「盡量」於星期日之前給你下星期的上課講義
- (你) 回家先看過 ...
- (一星期後 ...)
- (我) 會在上課中問 (你) 書中的內容
- (我) 會將課本中較難或較不清楚的地方再特別強調
- (你) 由於課前預習過，上課聽了就很有感覺，學習品質與效果自然非常好

本課程理想的進行方式 (II)

透過 laboratory exercise, (強迫) 養成課後馬上複習的好習慣

- (你) 上完課後馬上複習、消化 ...
- (18 小時後 ...)
- (TA and 我) 會在 lab 出兩題與上課內容非常相似的練習題，請 (你) 當場練習
- (你) 寫完就可以走了

本課程理想的進行方式 (III)

透過 homework、exam，深化 (你) 的理解，融會貫通所學到的知識，提升 (你) 的C++ programming 的程度

- (你) 上完Lab後開始做作業、約一星期以後繳交、深化你的理解
- (你) 考兩次期中考、一次期末考、融會貫通所學到的知識

- This is a **demanding** course but …

- Your efforts pay off.
- And the course should be a good one to take.